

Collaborative Evolution Strategies on Evolvable Hardware Networked Elements.

J.Vazquez, B.López, J.Valverde, E.de la Torre, T.Riesgo

I. INTRODUCTION

EVOLVABLE hardware (EH) systems are a paradigm which, although appeared some time ago, is nowadays acquiring an important momentum due to the feasibility of applying Dynamic and Partial Reconfiguration Techniques (DPR), available in modern FPGAs, as the native mechanism to allow such evolution. The advantages of achieving intrinsic evolution, that is, having the evolutionary loop implemented in the same device that will perform the operation, are extremely attractive: for instance, they permit autonomous, unattended adaptation to unknown problems, they theoretically allow to generate HW elements that perform the same functions as SW components, but with much higher performance or, even more attractive for highly-dependable systems, they may have self-healing capabilities.

An evolutionary loop consists of: 1) an evolutionary algorithm (EA) that proposes a candidate circuit, obtained by some transformations from previous candidate circuits by means of evolutionary operators such as mutation or crossover (understood, for EH terminology, as random variations of a circuit, or a combination of two circuits, respectively); 2) the newly generated circuit(s) are then obtained by means of DPR by differentially reprogramming the new circuit with respect to its parent(s) or brother(s); 3) evaluating the degree of adaptation of the circuit to the proposed system by means of the so-called fitness function

and, finally, 4) selecting the candidate(s) to become the parents of the next generation.

As it can be seen, a possible drawback of EH is evolution time. Typically, the EA execution time is negligible, as well as for the selection process. However, reconfiguration time and evolution time are key factors. Reconfiguration time is mostly technology dependent, while evolution time is application dependent. Evolvable Hardware based in DPR becomes attractive when these two factors become comparable. Some solutions provide smaller reconfiguration times by having Virtual Reconfigurable Circuits (VRC) which minimize reconfiguration time, but have much smaller scalability.

However, if EH components are networked, there becomes the possibility of using collaborative efforts among different nodes for accelerating evolution time. Evolution can be increased by parallelizing generations and evaluations, in order to reduce overall evaluation time. The representation of a circuit by its genome, as it will be seen, can be done with a very reduced amount of information, permitting to remotely exchange ‘individuals’ through the communication channels in a very inexpensive manner, giving chances for a successful collaborative evolution.

In this paper, some of these techniques are explored both in a SW model as well as in a real HW model implemented on top of an FPGA-based WSN node. Results will show which techniques result to be more appropriate, as well as evaluate the acceleration achieved by using the combined approaches.

II. STATE OF KNOWLEDGE

Parallelism in evolution strategies implemented on hardware gives rise to several approaches: it can be understood as a mere distribution of computing load or as a synergistic cooperative evolution among modules. Furthermore, an hypothetical imitation of natural co-evolution on hardware would represent another concept; not just a cooperation among equivalent modules to seek the best solution, but an evolution conditional on the role performed by the other individuals within the population.

As stated in [7], current parallel EAs have been approached mainly under three methodologies: master-slave, islands and cellular evolutionary algorithms (CEAs). In master-slave mode, the master node executes the algorithm and the evaluation task is distributed among individuals. In the island model there are several independent evolutionary processes (several islands), and the exchange of individuals

among them occurs at certain rate and frequency; the islands are panmictic, which means, all individuals are potential partners (random mating). In CEAs, the mating of individuals is restricted to a defined neighbourhood and the exchange frequency is usually high, as demanded by the cooperation strategy.

Among existing approaches to parallelize an ES as a mere distribution of processing, [1] performs an extrinsic evolution to design cellular automata (CA) rules by means of graphic processing units (GPUs); the decomposition of computing load among the threads can come from the number of cells within the CA, the number of training vectors or the number of individuals. GPUs are also employed in [4] to apply evolvable hardware to image filters based on Cartesian Genetic Programming; here, both task parallelism (evaluation of different chromosomes) and data parallelism (processing of different pixels) are exploited. Cellular Learning Automata based Evolutionary Computing has also been implemented on an FPGA [8], by designing a SIMD architecture, where the cells are the processing elements, arranged in a ring topology providing with local communication; the achieved high degree of parallelism is suitable for intrinsic hardware evolution.

Cooperation in [2] is applied to a compact genetic algorithm (GA) for EH, which is characterized by manipulating probability vectors instead of operating on the population; as for the cooperative evolution, each cell of a CA architecture works with its own sub-population and a “confident counter” is provided to point to the search direction. In contrast, the strategy in [3] for an equivalent scenario consists in partitioning the search space by splitting the solution vectors into smaller vectors. Another way to understand cooperation is through the bio-inspired concept of migration: in [5], a hierarchical structure for a distributed GA is implemented, so that migration exists towards upper levels; each instance represents a population, and periodically sends its best individuals to master instances. This feature represents a step forward in the adaptation and self-healing of EH systems, and is claimed in the article to resemble an artificial immune system. An implementation of such kind of hierarchical parallel EH on an FPGA is tackled in [9], where utilization of Network on Chip enables communication efficiency and scalability; a custom structure is developed to provide dynamic and partial reconfiguration (DPR) capability, and the parallel architecture consists of two levels: GA engines, which realize coarse-grain parallel GA with migration among subpopulations, and Target EHs, which act as slaves to the GA engines by distributing processing load. In [6] a software-simulated adaptive parallel genetic algorithm (PGA) is applied to the design of active filters; it applies an island model or coarse-grained PGA, where the population is divided into many subpopulations, each one carrying out its own evolutionary process, that exchange information between neighbours at certain time; maintenance of diversity and cost of communications are taken into consideration in the scheme of migration. Cellular-type cooperation is applied in [10] to implement a fault-tolerant VLSI architecture, which consists of two layers: the Computational Layer, in charge of the intended functionality (here GPS Attitude Determination System), and the Control Layer, that supervises the execution of the former one; both layers hold a fine-grained

PGA, with a large number of processing elements (PEs) that are operated by means of crossover with their immediate neighbours.

The presented approach introduces an island model of parallel evolution based on a network of processing nodes that can communicate wirelessly, enabling migration among populations. This contrasts to the consulted works, where the PGA is usually integrated into a single platform. Some exceptions include [11], where a large number of FPGAs are connected to a computer via optical fibers to study the dynamic self-organization of biological systems, by virtue of the programmable matter paradigm (although it does not explore cooperative evolution, but a very fine-grained modular co-evolution); or [12] and [13], multi-FPGA systems which are however integrated on a single circuit and communicate through a protocol implemented on the same device. The proposed cooperative scheme allows for the deployment of the processing nodes in a distributed environment; the elements are physically independent and communicate through a wireless protocol, which, in addition, permits heterogeneity of the devices. Finally, the proposed evolution strategy, based on a particular type of migration that preserves diversity, improves the convergence of the evolutionary algorithm, to a extent that depends on the exchange rate; this one can be tuned with a view to optimization.

III. EVOLVABLE HARDWARE PLATFORM

A. Background

The nodes that take part in the cooperative evolutionary process are custom Wireless Sensor nodes designed by the authors [14]; the node is meant to target high-performance applications with very competitive power consumption. High processing capabilities stem from the inclusion of specific hardware, here a low-power FPGA Spartan-6 LX150 by Xilinx. The node is provided with a modular architecture that comprises four layers: processing, communication, sensing and power supply, as shown in Figure 1. The FPGA included in the processing layer allows flexibility and DPR, while the communication layer enables establishing the migration scheme among nodes within the cooperative evolution.



Figure 1: Modular architecture of the Wireless Sensor node.

The evolvable hardware system was primarily implemented on a Virtex-5 FPGA [15], and targeted as demonstration the filtering of greyscale images with the single objective of quality of filtering, and afterwards migrated to the Spartan-6 LX150 FPGA [16] to perform a

multi-objective evolution taking power consumption into consideration.

The dynamically reconfigurable core of the system is a systolic array of PEs which process the pixels of the image through biaxial flow; given the set of 16 different arithmetical operations that the PEs perform, the evolutionary process implies modifying their combination according to a genetic algorithm. This modular DPR requires to extract and store the configuration file (partial bitstream) of every PE, in order to load them at run-time; as well, a common interface between PEs is required so that they are interchangeable (this interface is implemented as bus macros). On processing each single pixel, the genetic algorithm also determines, besides the functionality of the PEs, which inputs (out of the pixels belonging to the 3x3 window centred on the processed pixel) enter the systolic array, as well as the effective output that is taken from the array. Thus, online evolution provides flexibility and self-healing capabilities.

The application of a genetic strategy to evolvable hardware implies identifying which features or parameters of it act as genes, being their combination the so called chromosome. As well, the procedure to yield artificial evolution has to be specified – that is, how the offspring are generated –, and a mechanism of evaluation (fitness function) has to be defined so that individuals can be ranked, thus imitating natural selection. Here, the chromosome comprises the PEs, the selection of inputs and output and the filter's latency. The fitness function is defined as the mean absolute error between the pixels of the original image and the filtered one. As for the evolution strategy, in [15] offspring are generated from mutation of a single parent – selection $(1+\lambda)$ –, whereas in [16] descendent individuals are obtained by binary tournament and mutation within the parent population – selection $(\mu+\lambda)$.

B. Reconfigurable structure

The ability to perform DPR derives from the existence of the Internal Configuration Access Port (ICAP), which allows accessing the configuration memory from the FPGA itself. The way the configuration memory is organized depends on the morphology of the FPGA fabric. The Spartan-6 FPGA included in the node is divided into 12 clock regions, and each of them comprises 77 heterogeneous columns (CLB, IOB, DSP, RAM...). The minimum unit of the partial bitstream is the frame, which describes a fraction of a CLB column, so the height of the smallest module that can be reconfigured is the height of one clock region.

As described in [16], the PEs are designed with a height of one clock region and a width of 2 CLBs, with the Xilinx tools. Constraints for the location of the bus macros and the area to which the PEs are restricted are specified, and the routing of the elements is manually checked in order for the design to be reconfigurable. Finally, the partial bitstreams of the PEs are extracted and stored in the Flash memory of the board.

C. Architecture of the system

The architecture of the system, schematized in Figure 2, is divided into static and reconfigurable parts. The static region comprises the microprocessor (Microblaze), which

executes the evolutionary algorithm; the RAM blocks where images are stored; the Xilinx IP core to Access the ICAP (HWICAP) and a custom peripheral which manages the data flow of the filtering process, controls the activation of every sub-block, computes the fitness function, etc.

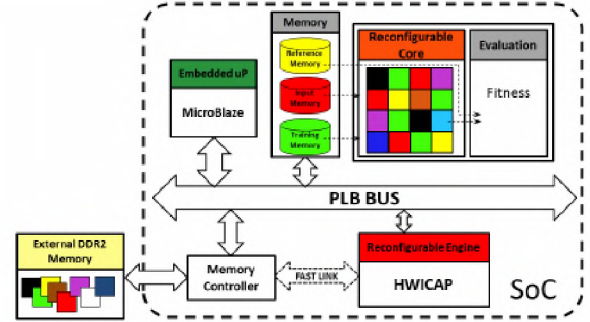


Figure 2: Architecture of the evolvable system

The design of the hardware is done with Xilinx XPS. Xilinx ISE and PlanAhead allow realizing the strategic layout of the system within the FPGA, and FPGA Editor enables supervising the reconfigurability of the design.

IV. COLLABORATIVE EVOLUTION STRATEGIES

As previously explained, an island model of parallel evolution has been developed within a network of processing nodes thanks to their capability of wireless communication. The evolution is based on a low power Spartan-6 FPGA connected by a serial port with the Wi-Fi communication module, enabling the information exchange and subsequently the migration among populations.

The system has been implemented on four nodes, being each one an island or population that takes part in the evolutionary process. These populations consist of four independent runs whose evolutions start at the same time with parents that are generated randomly. The four islands perform the same processing tasks but one of them serves as the master that starts the process and controls the migration scheme. This migration process is influenced by the exchange rate, defined as the inverse of the distance between one migration and the following. The results show that this exchange rate is an important factor whose incorrect choice could lead to poor results. For instance, a too high rate will derive in a loss of diversity and will increase the chances of stagnate in a local minimum, obtaining a worse fitness.

The migration process has been developed by a tournament method where the master island receives the best parent from each island and compares them choosing the one with the best fitness. The master asks each node for its best parent and keeps the information about its origin. The chosen parent replaces the worst in all the nodes but the origin node. The node that had the best parent is kept as originally and the worst parent is not replaced in order to preserve the diversity of the population having four different runs in the node. With this method, the possibilities of improvement of the best parent are increased without a high reduction in diversity due to the independence between evolvable runs within each island.

For an efficient exchange of parents, it is necessary to create a codification that reduces the amount of payload to be sent in each communication, resulting in a reduction of time and energy consumption. This is a key feature in a low energy consumption platform like the one used in this work. This efficient exchange is possible due to the storage of all the PEs in each node's memory together with a label which will be used in the following communications. This labelling method allows sending a PE using only 4 bits instead of 8 Kbyte needed to send the whole bitstream. This characteristic of the node allows for the scalability of the array without a significant increase in the need of information exchange. Another advantage is the possibility of working on the solution of different problems in only one node, toggling remotely between types of filtering almost immediately.

The payload is integrated by the number of rows and columns of the systolic array, the configuration of the multiplexers, the latency, the PEs configuration and the fitness. This last parameter is used as a comparison method between parents and, depending on the case, as a verification form, without the need of testing the sent parent in the master node. In table 1, a comparison between the sizes of the package depending on the size of the array is shown.

Size of the array	Package size
4x4	104 bits
6x6	200 bits
8x8	328 bits
12x12	680 bits

Table 1: Payload needed to be sent as a function of the size of the array

In spite of the reduced size of the information that is sent, the time required for this task is directly influenced by the communication mean with the Wi-Fi module. This transmission is done through a serial port and hence the time for sending each information package compared to the time required to configure the array or to filter the image is meaningful, as it is shown in Table 2.

Process	Time
Sending process	8.3ms
Reconfiguration of one PE	8ms
Filtering process	163 μ s

Table 2: Time employed to perform the different stages of the collaborative process

Therefore, the selection of the exchange rate only attends to fitness considerations, being the influence of exchange rate insignificant.

An additional advantage of the use of a simplified codification is that it allows for the use of different platforms in a unified way. The only requirement is that all the platforms have to contain a Wi-Fi module.

The collaborative distribution of tasks also makes the system robust and resistant to faults, because, in the event of one of the nodes becoming useless, the whole evolution would not be interrupted.

V. EXPERIMENTAL RESULTS

The system has been applied to obtain a filter of greyscale images with a salt-and-pepper noise of 5%. The same sample image has been used in all the processing islands in order to get a valid and fast fitness comparison among the different nodes.

Before the evolutionary strategy explained in the previous section, which produces the best fitness results, was found, several evolutionary schemes were tested.

In the first evolutionary scheme that was implemented in the system, the populations in each node were not independent and the descendants were chosen by means of a binary tournament within the parent population. Two among the four parents in the population were randomly selected and the best one was kept and used to obtain one descendant of the next generation. This method was inefficient, because the probability to generate a descendant from the worst member is null and from the best is 50%. With this method, in only one exchange, 50% of the evaluations were mutations of the best parent (the one chosen in the migration process).

Therefore, in order to improve the evolutionary strategy, the independence between the evolutions in the same node was mandatory. The tests that were carried out showed that this scheme with independent runs produces better results than the binary tournament process.

After fixing the evolutionary strategy, the number of descendants and parents per generation was established. Several tests with 1 and 2 descendants per generation were performed. Figures 1 and 2 show the results of the average fitness obtained in 200 evolutions from 160,000 evaluations each one. The different curves correspond to different exchange rates. As shown in Figures 1 and 2, it can be concluded that the evolution with two children per generation is a 15% better, regarding the asymptotic value of the curves. Only the last part of the curves is shown in these plots, because it is the relevant fragment to understand the results, being similar to an exponential function.

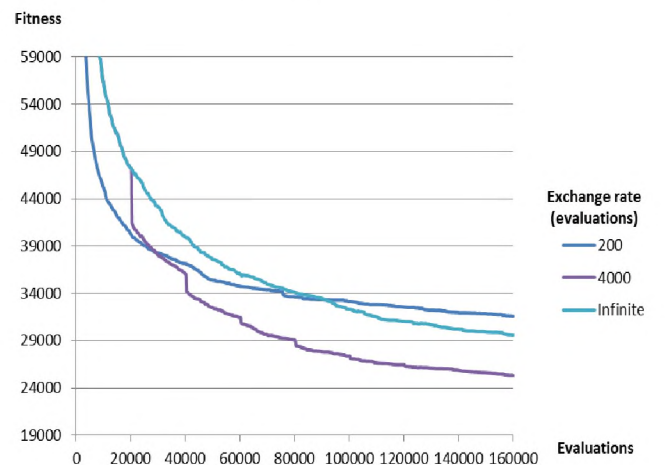


Figure 1: Fitness evolution by a collaborative process (4+1) applied to a greyscale image with 5% noise

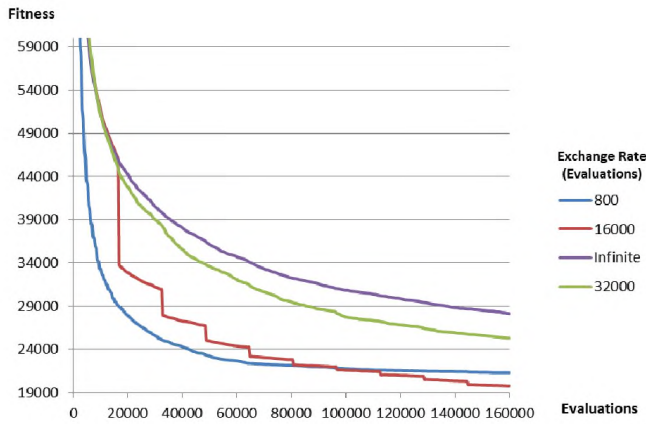


Figure 2: Fitness evolution by a collaborative process (4+2) applied to a greyscale image with 5% noise

Both tests prove the benefits of the exchange of information, proving, also that the capacity of avoiding local minimums is higher applying this collaborative scheme than with the conventional independent runs. Furthermore, it can be noted the influence of the exchange rate in the results, where the bests fitness corresponds to an average value of the exchange rate. This result can be explained as follows: a too high exchange rate could lead to an inevitable loss of diversity which would result in a reduction in the capacity of avoiding stagnation in the local minimums of the system.

On the other hand, although the improvement compared to the no-communication scheme is significant, a too low exchange rate does not take advantage of the whole potential of the evolvable system. At this regard, a minimum is obtained for the final fitness with an exchange rate close to 16.000 evaluations, which introduces an improvement of the 25% on the final fitness. Figure 3 shows the fitness evolution depending on the exchange rate.

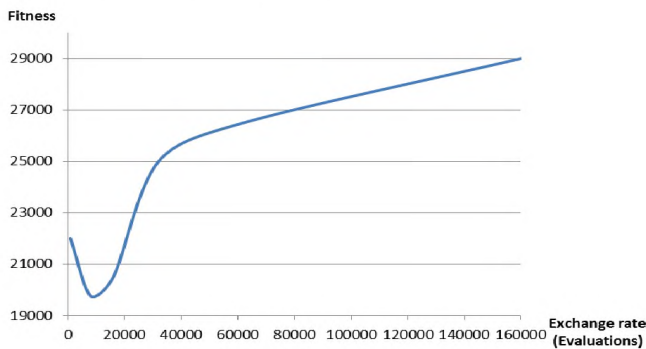


Figure 3: Fitness evolution depending on the exchange rate by a collaborative process (4+2) applied to a greyscale image.

The testing results prove that there is a high degree of dispersion but, in general terms, it can be stated that the collaborative scheme presents a higher resistance to local minimums and, therefore, produces better solutions.

Moreover, the time overhead associated with communication is not relevant due to the fact that exchanges take place only once every several generations, and the reduction in the achieved fitness is substantial.

VI. CONCLUSIONS

An evolvable system has been implemented in a network of processing nodes that can communicate wirelessly.

In contrast to other methods, the processing load is not only distributed but all the networked nodes work as independent islands which exchange their populations to obtain better solutions, unobtainable with the conventional strategies.

It has been applied to filter a greyscale image with salt-and-pepper noise and the strategies have been compared to each other and with the non-communicative model. Moreover, the influence of the distance between migrations has been analyzed, allowing coming up with optimal solutions.

Finally, a protocol of communication has been developed in order to reduce the amount of information needed to be sent and to minimize the time employed in this process.